PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 3/06

A1

(11) International Publication Number: WO 97/24655

(43) International Publication Date: 10 July 1997 (10.07.97)

(21) International Application Number: PCT/US96/19926

(22) International Filing Date: 27 December 1996 (27.12.96)

(30) Priority Data:
08/580,348 28 December 1995 (28.12.95) US

(71) Applicant: IPL SYSTEMS, INC. [US/US]; 124 Acton Street, Maynard, MA 01754 (US).

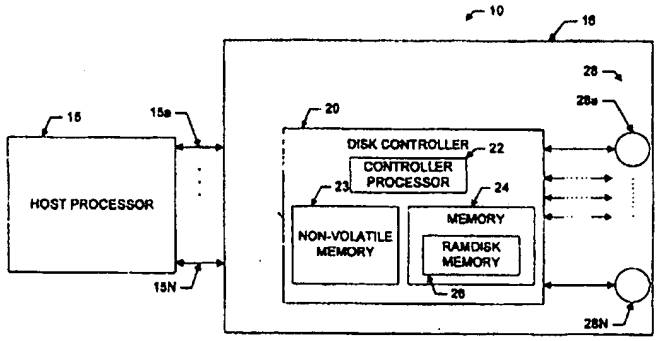
(72) Inventors: SCHARLAND, Michael, J.; 49 Long Avenue, Framingham, MA 01701 (US). GARBERO, Armando, D.; 12 Crane Road, Littleton, MA 01460 (US). IPPOLITO, Stephen, J.; 594 Old Marlboro Road, Concord, MA 01742 (US).

(74) Agent: KUDIRKA, Paul, E.; Bookstein & Kudirka, P.C., One Beacon Street, Boston, MA 02108 (US).

(81) Designated States: CA, JP, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: VIRTUAL RAMDISK



(57) Abstract

A disk controller (20) includes a controller processor (22) and a controller memory (24) having a plurality of memory locations with predetermined ones of the memory locations reserved for direct access by a host processor. With this particular arrangement, a disk controller (20) having a host accessible solid state memory at a much lower cost than a solid state disk is provided. The reserved memory locations of the controller memory may be referred to as a RAMDISK (26) which is configured such that it appears to the host processor as a solid state disk drive having a relatively small storage capacity. All read/write requests issued by the host processor and directed to the RAMDISK (26) are thus satisfied via reserved memory regions within a solid state controller memory (24) provided as part of a disk controller (20). Thus, the host processor need not read data from, or write data to, magnetic media such as a magnetic disk (28, 28a, 28n) drive memory or a magnetic tape memory. The net effect of providing a system having a RAMDISK (26) is similar to providing a disk storage system having a cache memory which operates with a one hundred percent cache hit rate on both read and write operations. The RAMDISK (26) may be provided as a software-selectable option within the disk controller (20), and, when enabled, the controller memory locations reserved for use as a RAMDISK (26) are allocated from available memory locations in the controller memory (24) identified from a poll memory available to the disk controller.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	**GB**	United Kingdom	**MW**	Malawi
AT	Austria	**GE**	Georgia	**MX**	Mexico
AU	Australia	**GN**	Guinea	**NE**	Niger
BB	Barbados	**GR**	Greece	**NL**	Netherlands
BE	Belgium	**HU**	Hungary	**NO**	Norway
BF	Burkina Faso	**IE**	Ireland	**NZ**	New Zealand
BG	Bulgaria	**IT**	Italy	**PL**	Poland
BJ	Benin	**JP**	Japan	**PT**	Portugal
BR	Brazil	**KE**	Kenya	**RO**	Romania
BY	Belarus	**KG**	Kyrgystan	**RU**	Russian Federation
CA	Canada	**KP**	Democratic People's Republic of Korea	**SD**	Sudan
CF	Central African Republic			**SE**	Sweden
CG	Congo	**KR**	Republic of Korea	**SG**	Singapore
CH	Switzerland	**KZ**	Kazakhstan	**SI**	Slovenia
CI	Côte d'Ivoire	**LI**	Liechtenstein	**SK**	Slovakia
CM	Cameroon	**LK**	Sri Lanka	**SN**	Senegal
CN	China	**LR**	Liberia	**SZ**	Swaziland
CS	Czechoslovakia	**LT**	Lithuania	**TD**	Chad
CZ	Czech Republic	**LU**	Luxembourg	**TG**	Togo
DE	Germany	**LV**	Latvia	**TJ**	Tajikistan
DK	Denmark	**MC**	Monaco	**TT**	Trinidad and Tobago
EE	Estonia	**MD**	Republic of Moldova	**UA**	Ukraine
ES	Spain	**MG**	Madagascar	**UG**	Uganda
FI	Finland	**ML**	Mali	**US**	United States of America
FR	France	**MN**	Mongolia	**UZ**	Uzbekistan
GA	Gabon	**MR**	Mauritania	**VN**	Viet Nam

VIRTUAL RAMDISK

FIELD OF THE INVENTION

This invention relates generally to disk storage systems and, more particularly, to disk controllers used in disk storage systems.

BACKGROUND OF THE INVENTION

As is known in the art, the increasing performance characteristics of central processor units (CPUs) and memories has not generally been matched by similar performance increases in input/out (I/O) systems. This mismatch has lead to the development of I/O systems which include arrays of magnetic disk memories. Such disk memory arrays may, for example, be implemented as a redundant array of inexpensive disks (RAID).

As is also known, a solid state disk is a solid state random access memory (RAM) that attaches to a host processor in a manner similar to that in which a magnetic hard disk drive attaches to a computer. The solid state disk appears to the host processor as a magnetic disk memory except that data access times and data transfer rates of the solid state disk are relatively fast compared with the access times and transfer rates of magnetic disk drive memories because solid state memories do not require any mechanical movement when performing read or write operations. One problem with solid state disks, however, is that they are relatively expensive compared with magnetic disk or magnetic tape storage devices. Thus, it would be prohibitively expensive to provide an array of solid state disks.

Magnetic disk drive memories may be provided having a solid state random access memory (RAM), generally referred to as a cache memory. Cache memories have a storage capacity which is relatively small compared with the storage capacity of a magnetic disk drive memory. Cache memories, on the other hand, have data access times which are relatively fast compared with the data access times of a disk drive. However, since only a portion of the total data stored on a magnetic disk drive memory is able to be stored in a cache memory, some type of scheme, such as a least recently used replacement (LRU) algorithm or a prefetch algorithm, is typically used to

determine what data to store in the cache memory.

The cache memory is transparent to the host processor except that, as mentioned above, when the data is read from or written to the cache memory, the access time is less than the access time associated with a magnetic disk drive. In this context, the term "transparent" means that the cache memory cannot be directly accessed by the host processor. Rather, the host processor issues a read or write request, and a disk controller or some other mechanism on a disk drive determines whether to satisfy the host request from the cache memory.

Several hard disk drives grouped together in a disk subsystem are typically referred to as a disk array. A disk array is intended to provide improved fault tolerance and performance compared tc several independent disk drives. The disk array is typically coupled to the host processor through a disk controller. The disk controller is coupled to the host processor through one or more host ports and to the hard disks of the disk array through one or more disk ports. Thus, the disks in the disk array do not connect directly to the host processor, but rather are coupled to the host processor through a disk controller.

The disk controller typically includes a controller processor which executes firmware to control the flow of data between the host processor and the array of disks. The controller may also include a controller memory which is typically provided as a solid state memory device and which can be used as cache memory to satisfy selected read and write requests directly in order to decrease the average data access time of a disk storage system. Such a cache memory, however, is transparent to the host processor.

The disk controller may allocate a portion of the cache memory to each of the disks in the disk array in accordance with a predetermined cache memory allocation scheme. For example, the disk controller may allocate equal memory portions of the cache memory to each of the disks in the disk array. Alternatively, the disk controller may permanently allocate portions of the cache memory to contain data stored on particular locations of one or more hard disks. This technique is generally referred to as

"pinning the cache" or "cache pinning." Thus, with a cache pinning technique, data stored in specific locations of the disk drives are always also stored in the cache memory.

In many computing environments, certain data processing applications, for example write-intensive data processing applications, could benefit greatly from having access to a relatively small amount of relatively fast memory since such write-intensive applications would not then have to wait for data to be written to the relatively slow disk drives of the disk storage system. Furthermore, if the disk storage system stores parity information in addition to data, then even more time is required to write the data to the disks. To take full advantage of a relatively fast storage medium, such as solid state RAM for example, the host processor should be able to store particular data directly on the relatively fast storage medium.

Since a cache memory of a disk controller is transparent to the host processor, it is not suited for storage of data in write-intensive data processing applications because the host processor cannot directly access the cache memory of a disk controller. Likewise, the pinned cache technique is not ideal, since it is typically very inconvenient for a host processor to store data in specific locations on a hard disk to thus insure that the data is subsequently available in the pinned cache memory. Solid state disks work well in write-intensive applications since a host computer can directly write data on the solid state disk, however, as mentioned above, a solid state disk is relatively expensive.

It would, therefore, be desirable to provide a system which is relatively inexpensive and which allows a host processor to directly store data in, and retrieve data from, a memory having a data access time which is relatively short compared with a data access time of a magnetic disk drive.

SUMMARY OF THE INVENTION

In accordance with the present invention, a disk controller includes a controller processor and a controller memory having a plurality of memory locations with predetermined ones of the memory locations reserved for direct access by a host

processor. With this particular arrangement, a disk controller having a host accessible solid state memory at a much lower cost than a solid state disk is provided.

The reserved memory locations of the controller memory may be referred to as a RAMDISK. The RAMDISK is configured such that it appears to the host processor as a solid state disk drive having a relatively small storage capacity. All read/write requests issued by the host processor and directed to the RAMDISK are thus satisfied via reserved memory regions within a solid state controller memory provided as part of a disk controller. Thus, the host processor need not read data from, or write data to, magnetic media such as a magnetic disk drive memory or a magnetic tape memory. The net effect of providing a system having a RAMDISK is similar to providing a disk storage system having a cache memory which operates with a one hundred percent cache hit rate on both read and write operations. It is important to note that the RAMDISK may be provided as a software-selectable option within the disk controller. If the RAMDISK option in the disk controller has been enabled, the controller memory locations reserved for use as a RAMDISK are allocated from available memory locations in the controller memory identified from a poll of memory available to the disk controller. It should also be noted that the size of the RAMDISK is also selectable and variable and is limited only by the amount of controller memory available to the disk controller.

The data written to the RAMDISK device should preferably persist across power cycles (i.e. between power up and power down cycles of the disk storage system). Since the RAMDISK is provided from memory locations of a volatile memory within the disk controller, the disk controller periodically transfers or "flushes" data stored in the RAMDISK to magnetic media for permanent storage. Such transfers are preferably done at a frequency sufficient to ensure that all data will be available from the magnetic media after a power down and subsequent powerup of the disk storage system. In one embodiment, such data flushes are performed in response to an idle timer providing a signal after a predetermined period of time or in response to a command issued by the host processor. The risk of losing data stored in the RAMDISK can be further reduced

by providing either a continuous power source to the disk controller or by equipping the disk controller with a battery of sufficient strength to maintain power to the disk controller for a relatively short period of time thereby allowing a memory flush to be performed in the event of an abnormal or forced powerdown condition of the disk controller or the disk storage system.

The data is flushed to a non-volatile disk or other non-volatile storage device rather than simply remaining in a battery powered volatile memory. Thus, with this flushing technique, the data is permanently stored and the data integrity of the system is thereby improved.

When flushed, the data may be spread evenly over all the disks in the disk array under control of the disk controller. This provides two advantages. First, standard parity protection schemes (e.g. RAID schemes) can be used to maximize the availability of the data. Second, the host processor system can view the disks in the disk array as still having a uniform storage capacity. Thus, mirroring or other techniques which require disks to have the same storage capacity may be used. Furthermore, by providing disks with uniform storage capacity, host configuration of the disks is relatively simple to accomplish.

When configured, the RAMDISK appears to the host processor merely as another logical unit (LU) within an existing disk storage system which LU is coupled to a disk controller. However, unlike stand alone solid state disk drives, no additional hardware or issues of host system addressability (I/O adaptor, bus or identification resources) arise by the addition of the RAMDISK. The RAMDISK is relatively easy to configure and implement in a disk storage system.

Cache memory schemes which make use of main storage inside the host processor system for I/O cache memories are generally limited by cost and the amount of available memory. By placing the RAMDISK storage external to the host processor, the amount of memory allocated to RAMDISK(s) within the system is both easily scaled and relatively inexpensive. Also, CPU processing overhead required for management of in-board caching schemes is eliminated.

Furthermore, in cache pinning techniques, it is desirable to control placement of data on a disk drive to ensure that the data is placed on a portion of the disk drive which is "pinned", thereby ensuring that the data will later be available in the cache memory. In many computer systems, however, control or visibility of data placement on a disk drive is limited or impossible and hence, "tuning" a cache memory to optimize performance (e.g. increase the number of cache hits) is relatively difficult.

In the present invention, however, the RAMDISK appears to the host processor as single disk drive having an assigned logical unit number and, thus, which is directly accessible by the host computer as if it were a disk drive in the disk array. By storing all data of interest in a single disk drive, the granularity necessary in controlling data placement increases and tuning is no longer required.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing features of the invention, as well as the invention itself may be more fully understood from the following detailed description of the drawings, in which:

FIG. 1 is block diagram of a computer system having a disk controller and a disk storage system coupled thereto;

FIG. 2 is a flow diagram illustrating the processing performed by a disk controller during an initialization procedure of a disk storage system;

FIG. 3 is a flow diagram illustrating the processing performed by a disk controller to process read and write requests from a host processor; and

FIG. 4 is a flow diagram illustrating the processing performed by a disk controller to determine whether all information presently stored in a memory of the controller should be transferred to a disk.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring now to FIG. 1, a computer system 10 includes a host processor 15 having a disk storage system 16 coupled thereto. Disk storage system 16 includes a disk controller 20 coupled to one or more of host ports 15a - 15N of host processor 15.

A disk array 28 having a plurality of magnetic disk drives 28a - 28N are coupled to the disk controller 20. Disk controller 20 includes a controller processor 22, a nonvolatile memory 23 and a solid state controller memory 24. Controller memory 24 may be provided, for example, as a dynamic random access memory (DRAM), a static RAM (SRAM) or any other type of memory. Controller memory 24 is preferably provided as a relatively inexpensive, relatively fast memory. A predetermined number of memory locations of controller memory 24 are reserved for direct access by host processor 15. The reserved memory locations of controller memory 24 may be referred to as a RAMDISK 26.

The RAMDISK 26 provides controller 20 having a memory region which appears to host processor 15 as a disk drive having a relatively small storage capacity. Thus, the memory locations of RAMDISK 26 are directly accessible by host processor 15.

Each read or write request issued by the host processor 15 and directed to RAMDISK 26 is satisfied by accessing solid state memory 24 within disk controller 20. Thus disk controller 20 need not access slower magnetic media such as a magnetic disk drives 28 or a magnetic tape memory. The net effect of providing controller 20 with RAMDISK 26 is quite similar to providing a controller having a caching disk device which operates with one hundred percent cache hit rates on both read and write operations.

It should be noted that the RAMDISK 26 may be provided as a software-selectable option within the disk controller 20. For example, disk controller 20 may be provided having a front panel which includes a display device (e.g. a liquid crystal diode (LCD)) and an input device. During an initialization procedure of the disk storage system 16, during a system power-up for example, controller initialization software may cause text to appear on the LCD display to query a user as to whether a RAMDISK option should be enabled. If a user indicates that the RAMDISK option should be enabled, then the user is further queried as to the desired size of the RAMDISK. As will be described in detail in conjunction with FIGs. 2 and 3, the controller processor 22 then performs those steps necessary to properly configure the RAMDISK with the host

processor 15. Thus, the portions of memory 24 used to provide RAMDISK 26 are allocated and designated as RAMDISK only if a RAMDISK option has been selected by a user or by the host processor.

As mentioned above, the size of RAMDISK 26 may be designated by a user via a front panel of a controller. Alternatively, the selection of a RAMDISK option and size of the RAMDISK may be specified by host processor 15. In this case a host processor interface protocol would be established to allow host processor selection of the RAMDISK option. Thus the size of RAMDISK 26 is software-selectable and is limited only by the amount of memory available to the disk controller 20.

In one embodiment, with the disk controller memory 24 having a memory size of about one-hundred and twenty-eight megabytes (MB) RAMDISK 26 may be designated as having a size typically in the range of about eight to thirty-two MB. In some applications however, it may be desirable or necessary to provide RAMDISK 26 having a size up to one-hundred and twenty-eight megabytes MB (i.e. the same size as controller memory 24). In this case, controller memory 24 would not be available for storage of data other than specified by host processor 15. In those applications where less than all of the controller memory 24 is specified as RAMDISK memory, those portions of memory 24 not reserved as RAMDISK 26 may be used for general purpose read/write cache, general purpose I/O cache for other I/O disk drives in the disk array 28 or for any other conventional use.

The actual memory locations used to provide RAMDISK 26 are allocated by first polling memory 24 to identify available memory locations in memory 24 and then allocating predetermined ones of the available memory locations as RAMDISK memory locations. The available memory locations of controller memory 24 reserved for RAMDISK operation are preferably contiguous memory locations but need not be contiguous memory locations. By providing RAMDISK 26 from contiguous memory locations of memory 24, mapping procedures for mapping memory locations and the data stored therein between RAMDISK 26 and disk drives 25 are simplified.

Disk controller 20 is provided having three operating modes which will be

described in detail below in conjunction with FIGs. 2-4. Suffice it here to say that disk controller 20 may operate in an initialization mode, a read/write request processing mode, and a data flush mode.

FIGs. 2 - 4 are a series flow diagrams showing the processing performed by disk controller 15. The rectangular elements (typified by element 42), herein denoted "processing blocks," represent computer software instructions or groups of instructions. The diamond shaped elements (typified by element 40), herein denoted "decision blocks," represent computer software instructions, or groups of instructions which affect the execution of the computer software instructions represented by the processing blocks. The flow diagrams do not depict syntax of any particular programming language. Rather, the flow diagrams illustrate the functional information one skilled in the art requires to generate computer software or firmware to perform the processing required of disk controller 20. It should be noted that many routine program elements, such as initialization of loops and variables and the use of temporary variables are not shown

Turning now to FIG. 2, during a system power up and subsequent disk array initialization procedure, as shown in decision block 40 controller processor first determines if a RAMDISK option has been selected and the RAMDISK size specified. If the RAMDISK option has not been selected then processing in the initialization mode concludes in step 50 as shown.

If the RAMDISK option has been selected the host processor is configured to allow direct access to the RAMDISK. Processing then continues to processing block 42 where an amount of memory equal to a requested RAMDISK memory capacity is reserved from an available memory pool of a controller memory. The reserved memory will be set aside for RAMDISK use while a disk array is coupled to the disk controller is operational. Provisions for the case where sufficient memory is not available are implementation-specific and will not be discussed here in detail. Suffice it to say that if a case arises where sufficient memory is not available for a RAMDISK of a desired size, then the host processor is informed that insufficient memory is not available and no

RAMDISK memory is assigned.

Once memory locations have been reserved for RAMDISK use, then as shown in block 44, a host image for the actual disk drives in the disk array is altered to show a reduction in available memory capacity. The term "host image" refers to the appearance to the host computer of the disk drives in the disk array. That is, host image refers to the appearance the disk controller presents to the host processor of the disk drives in the disk array such as disk capacity, physical disk characteristics and any other parameters of the disk drives in the disk array .

For example, if the disk array is provided having eight disk drives each having two gigabytes (GB) of storage capacity, the disk controller could provide this information to the host processor. However, the disk controller could alternatively represent to the host processor that the disk array included sixteen disk drives with each of the sixteen disk drives having a one GB storage capacity. In this case, the disk controller would be required to perform some logical to physical mapping between the host processor and the disk array in order to allow the host processor to properly access each of the eight disk drives. The particular image the disk controller presents to the host processor depends at least in part on what processing is normally performed by the disk controller.

In the case where a RAMDISK is selected and configured in the host processor regardless of the amount actual physical memory available on each of the disk drives, a portion of the disk drive memory must be reserved for storage of data from the RAMDISK. Thus, the image provided to the host with respect to the amount of memory available in a disk drive must be reduced by at least the amount of memory reserved to accept data from the RAMDISK.

In one embodiment the amount of memory capacity by which the disk drives must be reduced corresponds to the amount of memory space allocated on each disk drive for the non-volatile storage of RAMDISK data and may be computed as:

$$SIZE_{NEW} = SIZE_{ORIGINAL} - M/D$$

in which:

$SIZE_{NEW}$ corresponds to the apparent memory size of the disk drives to the host processor after allocation of the RAMDISK memory;

$SIZE_{ORIGINAL}$ corresponds to the actual memory size of the disk drives;

M corresponds to the configured memory size of the RAMDISK memory; and

D corresponds to the number of disk drives in the disk array.

Alternatively it may be desirable to simple reserve an amount of memory in each disk drive which is equal to the amount of memory in the entire RAMDISK.

As is shown in processing block 46, the configured RAMDISK memory capacity is divided equally among all of the disk drives in the disk array. Each disk drive must be accessed during the initialization process to retrieve data stored in the memory portion of the disk drive reserved for storage of RAMDISK data. The data is retrieved from the reserved memory area of the magnetic disk drive and stored in the reserved memory area of the controller memory reserved as the RAMDISK. Thus during subsequent read operations of the host processor data may be retrieved from the RAMDISK and data from write operations of the host processor will be stored and/or updated in the RAMDISK and then flushed to the disk drives.

It should be noted that if, upon power up of the controller, the information stored in the nonvolatile memory of the controller indicates that a RAMDISK was configured during a previous power up, then this indicates that the areas of the disk drives reserved for storage of RAMDISK data may contain data during the last operating period of the system. Thus, the data is read from the reserved areas of magnetic disk drives back into the RAMDISK.

If the power up of the processor is the initial power up of the controller and the RAMDISK option is selected, then the controller still reads the reserved areas of the magnetic disk drive memories, however, such memory areas will not contain any useful data. The host processor should know that this RAMDISK has never been used before and thus the host processor may want to initiate write operations to the RAMDISK rather than read operations since no meaningful data would yet be stored in the RAMDISK.

Host processor and disk controller response to a case where a data retrieval operation fails is implementation-specific. A data retrieval operation may fail due to failure of one or more disk drives in the disk array upon power up or due to physical removal of a disk drive from the disk array.

Before exiting the initialization routine, as shown in step 48, an idle timer flag is reset and a dirty flag is set to a logical FALSE value. The idle timer flag and dirty flag control a flush process by detecting appropriate points in time at which to perform RAMDISK flush operations based on activity and validity of data store in the RAMDISK.

Specifically, the idle timer counts for a predetermined period of time and provides a signal which indicates whether data in the RAMDISK has been written to the disk array within that predetermined period of time. The dirty flag indicates whether data stored in the RAMDISK has been modified since it was last written to the disk array. Thus the dirty flags indicate whether data stored in the disk array is valid (i.e. up to date).

The dirty flags can be used to track the validity of RAMDISK data in varying granularities. Choice of this granularity can be made based on RAMDISK size, size of the smallest modifiable unit of storage, amount of available memory, or other such criteria.

In one embodiment, the disk controller can support 500 MB of cache, however, a relatively small amount of memory from the controller memory is reserved for use as a RAMDISK. For example the RAMDISK may have a storage capacity in the range of about 8-32 MB. Thus in this embodiment only a single dirty flag is required for the entire RAMDISK.

Thus, in the case, where only a relatively small amount of controller memory is reserved for the RAMDISK; a single dirty flag may be used to indicate whether any write operations have occurred to any memory locations of the RAMDISK. The use of a single dirty bit works satisfactorily for a RAMDISK provided from a relatively small amount of memory because flushing of the RAMDISK can be done relatively quickly.

If, however, the RAMDISK was relatively large or it was desired to track the

validity of data in a smaller memory size than the entire RAMDISK, a plurality of dirty flags could be used. For example, if the RAMDISK were 32 MB of memory then it may be desirable to use four dirty flags with each of the four dirty flags representing an 8 MB portion of the 32 MB RAMDISK. Alternatively still, a dirty bit could be assigned for every 100 bytes of RAMDISK memory. Such an approach, however, would lead to a plurality of dirty bits and a concomitant complexity in tracking dirty bits and multiple flushing operations.

FIG. 3 shows processing performed by disk controller 20 while processing read and write requests from the host processor 15.

As shown in decision block 52, controller processor first determines if a read or write request has been received from host processor 15. When the disk controller detects a read or write request, processing continues to block 54 where the controller processor determines if the host processor request is directed to the RAMDISK.

If the host request is not directed to the RAMDISK, then processing continues to processing block 56 where the request is processed for the appropriate disk drive in the disk array.

If the host request is directed to the RAMDISK, processing continues to processing block 58 where the controller processor performs a mapping step to translate the data being addressed by the host processor into a corresponding location within the reserved memory area of the controller memory 24 (FIG. 1). This mapping function is fairly implementation-specific, with the only requirement being that there exist a unique storage location for each host-addressable element of the RAMDISK.

For example, if the RAMDISK appears to the host processor as a block-oriented disk device having 512 bytes per block, when the host processor accesses logical block address (LBA) 0, data from bytes 0 through 511 of the reserved memory area (i.e. the RAMDISK) are read and written as necessary. When the host processor accesses LBA 1, bytes 512 through 1023 of the RAMDISK are affected, access to LBA 2 affects bytes 1024 through 1535 of the RAMDISK and so on. Thus in this example, the host processor will view the RAMDISK as having a maximum LBA of:

LBA_{MAX} = (Configured RAMDISK memory size / bytes per block) - 1

Alternatively, if the RAMDISK appears to the host processor as a track-oriented disk device having 32 kilobytes (KB) per track, and the RAMDISK is not provided from contiguous memory locations of the controller memory but rather is provided from a plurality of 32 KB buffers from various memory locations of the controller memory, then in this case a pointer table entry should be generated which contains the address of a start buffer. An offset within the pointer table can then be defined as the host track number. Therefore, when the host access a particular track (e.g track number N), the memory pointed to by table entry number N is affected.

Next as shown in decision block 60 the controller processor determines the type of request issued by the host processor. If decision is made that the host request corresponds to a write request, then processing continues to blocks 62 and 64 where the host data is received into a reserved memory area of a controller memory (i.e. the RAMDISK) and a dirty flag is set to a logical TRUE value. Thus, in response to a write request, the disk controller accepts data from the host into the appropriate memory area, and additionally identifies the memory area as dirty so that data stored in the memory area will be flushed to the sik array at an appropriate point in time. It should be noted that data becomes valid when it is written to RAMDISK. The data does not have to be written to the disk drives to be considered valid by the host processor.

If decision is made that the host request corresponds to a read request, then processing continues to block 66 where the requested data is immediately returned to the host processor from the RAMDISK. Thus in response to a host read request, the disk controller is able to immediately return the requested data to the host processor from the appropriate locations within the reserved memory area of the controller memory

Finally, regardless of whether the host request was a read request or a write request, processing continues to processing block 68 where an idle timer is reset to a predetermined timeout value. After the idle timer is set, the controller processor subsequently resumes checking for idle periods to flush data if necessary.

Referring now to FIG. 4, the processing steps performed in disk controller 15 to perform a flush operation are shown. In decision block 70, decision is made as to whether a host processor directed flush was received. A host processor directed flush is the highest priority flush request (a possible exception to this is a system which has been equipped with a short-term battery to allow flushing in the event of a power failure). Host-directed flushing provides the host processor with the ability to force all RAMDISK data to be backed up at logical boundaries within the host processing sequence (e.g. dismounting a database). Thus, a host ordered flush typically occurs when the host processor determines that it will no longer access data stored in the RAMDISK.

For example, if RAMDISK is being used in some particular database application, and a user finishes with the database and is logically dismounting the database from the host processor, then the host processor would determined that this was an appropriate time to perform a host flush operation. Thus if decision is made that a host directed flush was received, processing flows directly to processing block 78 where a predetermined amount of data stored in the RAMDISK memory is transferred to each disk drive in the disk array.

The predetermined amount of data is preferably of a size corresponding to M/D bytes where M is the configured size of the RAMDISK and D is the number of disk drives in the disk array. Processing then flows to processing block 80 where the dirty flag is set to a logical value of FALSE and processing again begins at decision block 70.

If decision is made in decision block 70 that a host directed flush has not been received, then processing continues to decision block 72. Decision block 72 checks to see if an idle timer has expired. A flush request having a lower-priority than a host flush request results from the expiration of the idle timer. Expiration of the idle time which results in an idle timer flush request indicates that the host processor has not addressed the RAMDISK for a predetermined period of time.

The idle time period for flushing may be selected in accordance with a plurality of

factors and a preferred idle time period may be different in different applications. For example, the idle time period may be select as one minute. This time period may be selected because typical computer systems do not shut down in less than one minute and battery back-up systems typically sustain disk subsystems for a period of time which is not less than one minute. However, an idle time may also be selected based on performance of the computer system. For example, if a system performs many write operations every thirty seconds it may be desirable to set the idle time to thirty seconds.

As shown in blocks 72, 74 when a timeout occurs, the idle timer is immediately restarted and the dirty flags are scanned to determine whether any portions of the RAMDISK data in reserved memory are out of sync with the non-volatile copy of the RAMDISK data stored on the disk drives of the disk array. If any such portions are found, a flush operation will proceed, else the timeout condition will be ignored.

If the flush proceeds, as is always the case in a host-directed request, all dirty RAMDISK memory locations (e.g. RAMDISK blocks) which are indicated as being "dirty" will be written from the reserved memory area of the controller memory (i.e. the RAMDISK) to the appropriate reserved areas on each of the disk drives in the disk array. After flushing is complete, the dirty flags are reset and normal RAMDISK processing of read and write requests. Provisions for the case where a flush fails are once again implementation-specific.

Having described preferred embodiments of the invention, it will now become apparent to one of ordinary skill in the art that other embodiments incorporating their concepts may be used. It is felt therefore that these embodiments should not be limited to disclosed embodiments, but rather should be limited only by the spirit and scope of the appended claims.

What is claimed is:

CLAIMS

1. A disk controller, coupled between a host processor and a disk drive, the disk controller comprising:

a controller processor;

a controller memory having a first plurality of memory locations wherein predetermined ones of the plurality of memory locations of said controller memory are directly accessible by the host processor; and

means for providing a disk image to the host processor wherein the disk image is determined based on the predetermined ones of the plurality of memory locations of said controller memory are directly accessible by the host processor.

2. The disk controller of claim 1 further comprises means for selecting the number of predetermined memory locations directly accessible by the host processor.

3. The disk controller of claim 2 further comprises:

means for identifying available memory locations in said controller memory; and

means for allocating the predetermined memory locations of said controller memory from the available memory locations identified by said means for identifying such that the host processor can directly access the predetermined memory locations.

4. The disk controller of claim 3 wherein said means for identifying includes means for polling said controller memory to identify available memory locations.

5. A method for initializing a disk storage system having a disk controller coupled to a plurality of disk drives, the method comprising the steps of:

(a) determining if predetermined memory locations of a controller memory have ben reserved for direct access by a host processor;

(b) in response to said determining step indicating that predetermined memory locations have been reserved for direct access by the host processor, configuring the predetermined memory locations for direct access by the host processor;

(c) allocating the predetermined memory locations from the controller memory;

(d) modifying disk drive parameters stored in a host processor to reflect that at least one of the plurality of disk drives coupled to the disk controller has a reduced memory capacity; and

(e) reserving a portion of disk drive memory in at least one the plurality of disk drives, wherein the reserved portion of disk drive memory has a memory size sufficient corresponding to a portion of the predetermined memory locations of the controller memory configured for direct access by the hots processor.

6. The method of claim 5 wherein the step of reserving a portion of disk drive memory includes the step of reserving disk drive memory space on each of the plurality of disk drives by an amount equal to:

$$SIZE_{NEW} = SIZE_{ORIGINAL} - M/D$$

in which:

$SIZE_{NEW}$ corresponds to a number equal to the apparent memory size of each of the plurality of disk drives as the plurality of disk drives appear to the host processor after the modifying step;

$SIZE_{ORIGINAL}$ corresponds to a number equal to the actual memory size of each of the plurality of disk drives;

M corresponds to a number equal to the configured memory size of the predetermined memory locations configured in said configuring step; and

D corresponds to a number equal to the plurality of disk drives.

7. The method of claim 6 further comprising the steps of:

accessing each of the plurality of disk drives;

retrieving data stored in the reserved memory portion of the disk drive; and

storing data retrieved in the retrieving step in the predetermined memory locations of the controller memory configured for direct access by the host computer.

8. The method of claim 7 further comprising the step of setting a first logical switch to a first value which indicates that a predetermined period of time has not elapsed.

9. The method of claim 8 further comprising the step of setting a second logical switch to a first value which indicates that data stored in the predetermined memory locations of the controller memory has not been changed since the data was last written to one or more of the plurality of disk drives.

10. In a disk storage system having a disk controller coupled to a plurality of disk drives, a method of processing a request issued by a host processor, the method comprising the steps of:

(a) determining if the request is directed to a reserved memory region of a controller memory directly accessible by the host processor;

(b) in response to the request being directed to the reserved memory region of the controller memory, mapping a host processor issued address into a corresponding location within the reserved memory area of the controller memory; and

(c) determining the type of request issued by the host processor.

11. The method of claim 10 wherein in response to determining in said determining step that the request is a write request further then performing the steps of:

receiving data into the reserved memory area of the controller memory;

and

setting a first logical flag to a first value indicating that data has been written into the reserved memory area of the controller memory.

12. The method of claim 10 wherein in response to determining in said determining step that the request is a read request then performing the steps of:

retrieving data stored in the reserved memory area of the controller memory; and

providing data retrieved in said retrieving step to the host processor.

13. The method of claim 10 wherein the mapping step includes the step of:

accessing, via the host processor, a first logical block address to access data in a first sequential set of addresses of the reserved memory area; and

accessing, via the host processor, a second logical block address to access data in a second sequential set of addresses of the reserved memory area wherein the first and second logical block addresses are sequential and the corresponding first and second set of addresses of the reserved memory area are sequential.

14. The method of claim 10 wherein the mapping step includes the step of:

generating a pointer table entry which contains an address of a start buffer;

generating an offset value within the pointer table which defines a host track number; and

accessing, via the host processor, a particular track of a host accessible logical unit by specifying a particular offset value within the pointer table.

15. A method of flushing data from a controller memory to a plurality of disk drives comprising the steps of:

(a) transferring from a reserved memory portion of the controller memory to a first

one of the plurality of disk drives, a predetermined amount of data stored in a first M/D memory locations of the reserved memory portion of the controller memory where M corresponds to a configured size of a reserved memory portion of the controller memory and D corresponds to a number equal to the number of disk drives in the plurality of disk drives;

(b) transferring from a reserved memory portion of the controller memory to a second one of the plurality of disk drives, a predetermined amount of data stored in a second M/D memory locations of the reserved memory portion of the controller memory where M corresponds to a configured size of a reserved memory portion of the controller memory and D corresponds to a number equal to the number of disk drives in the plurality of disk drives; and

(c) setting a first logical switch to a first logical value indicating that no data has been written to the reserved memory region of the controller memory since data was last transferred from the reserved memory region of the controller memory to the plurality of disk drives.

16. The method of claim 15 further comprising the step of:

determining whether a predetermined period of time has passed since the last time a host processor addressed the reserved memory regions of the controller memory; and

in response to determining that the predetermined amount of time has passed, determining whether any data stored in the reserved memory portion of the controller memory is out of sync with the corresponding data stored on the plurality of disk drives.

17. A disk storage system coupled to a host processor, the disk storage system comprising:

(a) a plurality of disk drives;

(b) a disk controller coupled to the plurality of disk drives the disk controller

comprising:

a controller processor;

a controller memory;

means for determining if predetermined memory locations of the controller memory have been reserved for direct access by a host processor;

means for indicating that predetermined memory locations have been reserved;

means for configuring the predetermined memory locations for direct access by the host processor;

means for allocating the predetermined memory locations from the controller memory;

means for modifying an apparent disk size viewed by the host processor to reflect that at least one of said plurality of disk drives coupled to said disk controller has a memory capacity less than an actual memory capacity of said at least one disk drive; and

means for reserving a portion of disk drive memory in at least one of said plurality of disk drives, wherein the reserved portion of disk drive memory has a memory size corresponding to a portion of the predetermined memory locations of said controller memory.

18. The apparatus of claim 17 wherein said means for reserving a portion of disk drive memory includes means for reserving disk drive memory space on each of the plurality of disk drives by an amount equal to:

$$SIZE_{NEW} = SIZE_{ORIGINAL} - M/D$$

in which:

$SIZE_{NEW}$ corresponds to a number equal to the memory size of each of said plurality of disk drives as said plurality of disk drives appear to the host processor after said means for modifying modifies the apparent disk size;

$SIZE_{ORIGINAL}$ corresponds to a number equal to the actual memory size of

each of said plurality of disk drives;

M corresponds to a number equal to the configured memory size of the memory configured in said configuring step; and

D corresponds to a number equal to said plurality of disk drives coupled to said disk controller.

19. The apparatus of claim 19 further comprising:

means for accessing each of said plurality of disk drives;

means for retrieving data stored in the disk drive memory space reserved by said means for reserving; and

means for storing data retrieved by said means for retrieving in the predetermined memory locations of said controller memory reserved for direct access by the host processor.

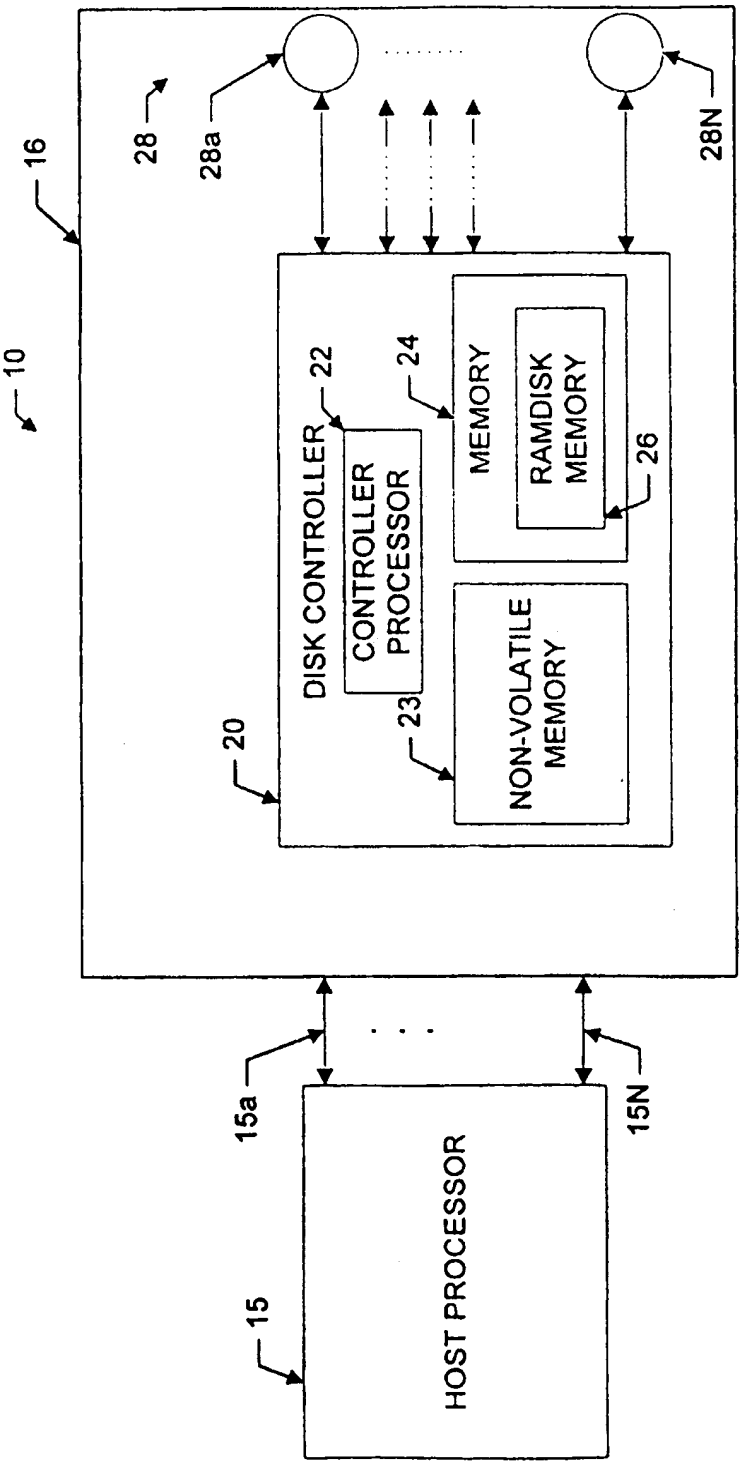

10
16
28
28a
28N
DISK CONTROLLER
22
CONTROLLER PROCESSOR
23
24
MEMORY
RAMDISK MEMORY
26
NON-VOLATILE MEMORY
20
15a
15N
HOST PROCESSOR
15

FIG. 1

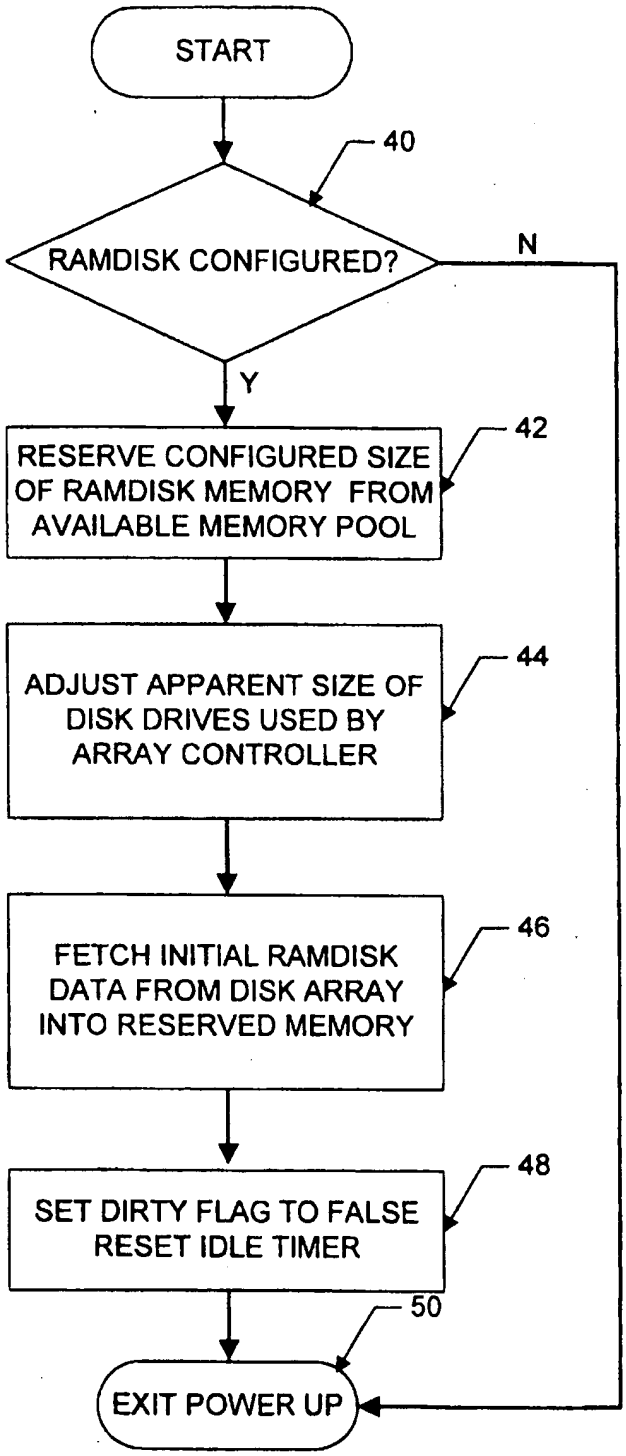

START
40
RAMDISK CONFIGURED?
N
Y
42
RESERVE CONFIGURED SIZE OF RAMDISK MEMORY FROM AVAILABLE MEMORY POOL
44
ADJUST APPARENT SIZE OF DISK DRIVES USED BY ARRAY CONTROLLER
46
FETCH INITIAL RAMDISK DATA FROM DISK ARRAY INTO RESERVED MEMORY
48
SET DIRTY FLAG TO FALSE RESET IDLE TIMER
50
EXIT POWER UP

FIG. 2

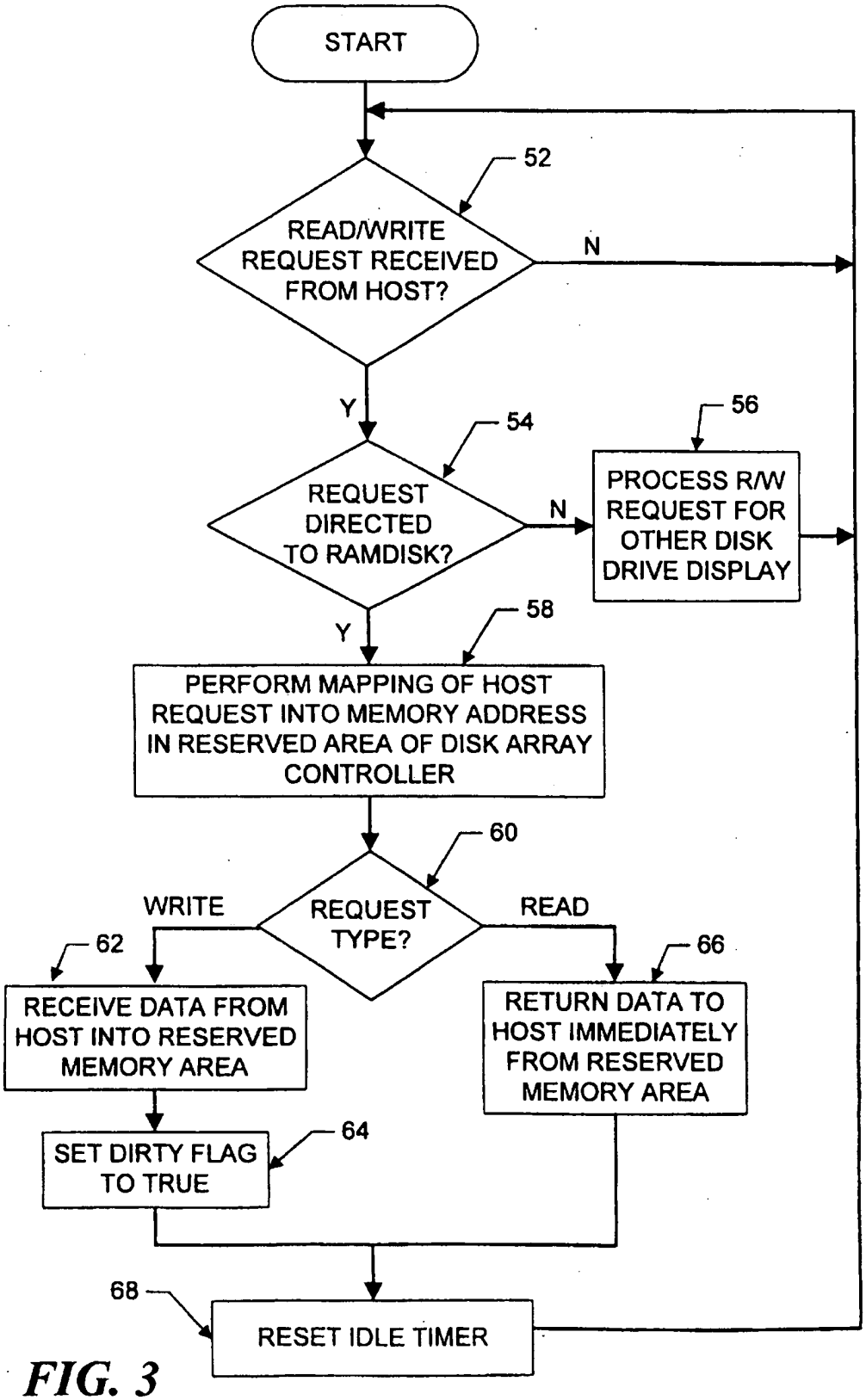

START
52
READ/WRITE REQUEST RECEIVED FROM HOST?
N
Y
54
REQUEST DIRECTED TO RAMDISK?
N
56
PROCESS R/W REQUEST FOR OTHER DISK DRIVE DISPLAY
Y
58
PERFORM MAPPING OF HOST REQUEST INTO MEMORY ADDRESS IN RESERVED AREA OF DISK ARRAY CONTROLLER
60
REQUEST TYPE?
WRITE
READ
62
RECEIVE DATA FROM HOST INTO RESERVED MEMORY AREA
66
RETURN DATA TO HOST IMMEDIATELY FROM RESERVED MEMORY AREA
64
SET DIRTY FLAG TO TRUE
68
RESET IDLE TIMER

FIG. 3

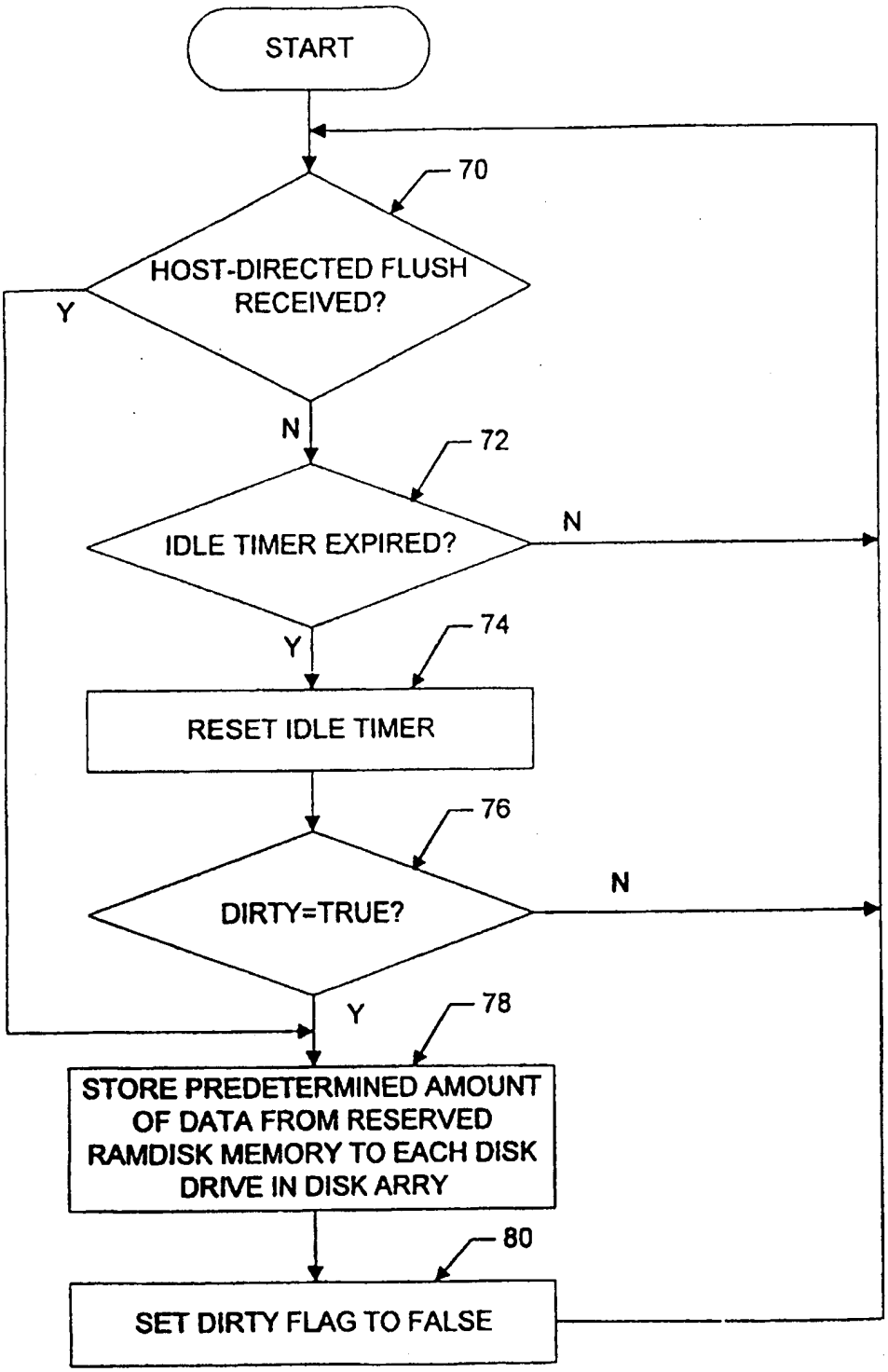

START
70
HOST-DIRECTED FLUSH RECEIVED?
Y
N
72
IDLE TIMER EXPIRED?
N
Y
74
RESET IDLE TIMER
76
DIRTY=TRUE?
N
Y
78
STORE PREDETERMINED AMOUNT OF DATA FROM RESERVED RAMDISK MEMORY TO EACH DISK DRIVE IN DISK ARRY
80
SET DIRTY FLAG TO FALSE

FIG. 4

INTERNATIONAL SEARCH REPORT

Inten ..al Application No
PCT/US 96/19926

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F3/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	PATENT ABSTRACTS OF JAPAN vol. 012, no. 136 (P-694), 26 April 1988 & JP 62 257553 A (TOSHIBA CORP), 10 November 1987,	1
Y		10
A		5,17

X	PATENT ABSTRACTS OF JAPAN vol. 011, no. 015 (P-536), 16 January 1987 & JP 61 190644 A (TOSHIBA CORP), 25 August 1986,	1,2
A	see abstract	5,10,17

Y	EP 0 564 699 A (FUJITSU LTD) 13 October 1993	10
A	see abstract; claims 1,2,4; figures see column 4, line 10 - line 17	1,2,5, 15,17

	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

1

Date of the actual completion of the international search	Date of mailing of the international search report
7 April 1997	24.04.97

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (- 31-70) 340-2040, Tx. 31 651 epo nl, Fax (- 31-70) 340-3016	Durand, J

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inter nal Application No
PCT/US 96/19926

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	IBM TECHNICAL DISCLOSURE BULLETIN, vol. 26, no. 4, September 1983, NEW YORK US, pages 1855-1857, XP002028955 BAKER E.D. ET AL.: "Electronic diskette unit" see the whole document	13
A	ICL TECHNICAL JOURNAL, vol. 8, no. 2, 1 November 1992, pages 332-346, XP000320465 JENKINS A W: "ESS - A SOLID STATE DISC SYSTEM FOR ICL SERIES 39 MAINFRAMES" see page 342, paragraph 2 - paragraph 3; figure 6	14
A	PATENT ABSTRACTS OF JAPAN vol. 014, no. 199 (P-1040), 23 April 1990 & JP 02 039342 A (FUJITSU LTD), 8 February 1990, see abstract	15
A	DE 295 12 593 U (FRANCK PETER HEINZ) 12 October 1995 see page 5, paragraph 2; figure 1	15

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inter. nal Application No
PCT/US 96/19926

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0564699 A	13-10-93	US 5420998 A	30-05-95
		JP 6083708 A	25-03-94
DE 29512593 U	12-10-95	DE 19532147 A	06-02-97